

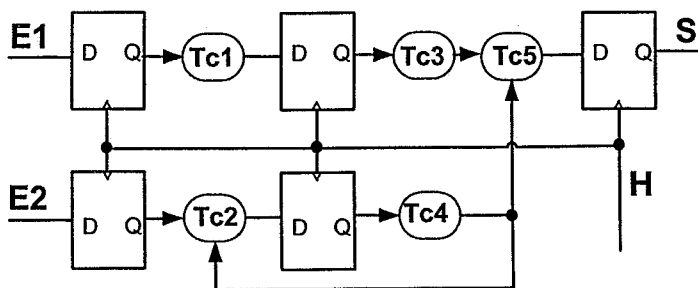
**Exercice 2 :**

On souhaite transmettre des données d'un ordinateur A vers un ordinateur B, au travers d'une ligne série en respectant le protocole RS232 (1 bit de Start, 8 bits de data et 1 bit de Stop). Le ordinateur A doit envoyer au ordinateur B  $x$  octets de données toutes les 5 ms. La vitesse de transmission de cette ligne série est de  $V$  bits/s.

- 1) Donner l'expression du temps de transmission des  $x$  octets.
- 2) Dédire l'inégalité nécessaire pour respecter le cahier des charges.
- 3) Application numérique : si  $x = 64$  et  $V = 28800$ , est ce que le cahier des charges est respecté ?

**Exercice 3 :**

Soit le schéma logique de la figure ci-contre: On supposera que les bascules ont un temps de SETUP, de HOLD et de PROPAGATION non nuls que l'on notera (respectivement)  $T_{su} = 2ns$ ,  $T_h = 2ns$  et  $T_{pd} = 4ns$ .



- 1) Quelle est la fréquence maximale de l'horloge H permettant un bon fonctionnement du montage ? Justifiez votre réponse.

Temps de propagations des blocs combinatoires:  $16ns \leq T_{c1} \leq 19ns$ ,  $14ns \leq T_{c2} \leq 16ns$ ,  $9ns \leq T_{c3} \leq 10ns$ ,  $4ns \leq T_{c4} \leq 5ns$ ,  $6ns \leq T_{c5} \leq 10ns$ .

**Exercice 4 :**

On considère une carte microcontrôleur à base d'un Microcontrôleur 80C592. On souhaite adjoindre à la carte un circuit de Conversion Numérique->Analogique (CNA). Ce CNA permet de produire une tension analogique comprise entre 1 et 4 volts à partir d'une donnée numérique (non signée) sur 8 bits.

- 1) Donner la résolution en sortie du CNA (la plus petite variation de tension pouvant être générée).

Ce CNA dispose des signaux suivants :

- 8 bits de données D7 à D0.
- 1 bit  $\overline{CS}$  pour activer le boîtier.
- 1 signal de sortie analogique.
- VCC et GND, les deux signaux d'alimentation du circuit.

La carte microcontrôleur dispose d'un certain nombre de circuits annexes occupants le plan mémoire externe du microprocesseur. Ce plan d'organisation de la mémoire externe est le suivant :

0000H à 6FFFH	Mémoire RAM externe RAM1
8000H à AFFFH	Circuit annexe d'affichage vidéo
B000H à EFFFH	Circuit d'acquisition de données RDS

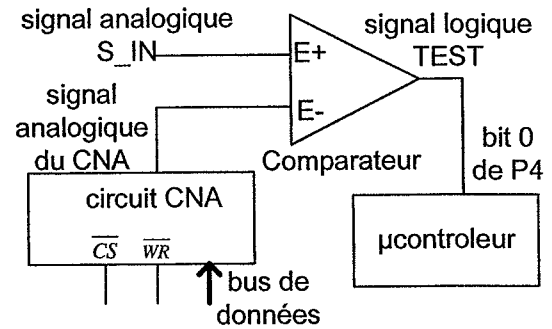
Le reste de la mémoire est considérée comme libre.

- 2) Proposer une zone mémoire libre pour implanter le circuit CNA. On considère que ce circuit pourra occuper au plus une page de 256 octets pour simplifier les équations de décodage d'adresses, dans ce cas quels sont les bits d'adresse intervenant dans le décodage?
- 3) Donner l'équation du bit de sélection du boîtier CNA et proposer un schéma d'implantation.

On souhaite réaliser un convertisseur analogique/numérique par logiciel en utilisant le CNA.

On réalise pour cela le montage ci contre.

Le principe de la conversion analogique/numérique consiste à contrôler le circuit CNA de sorte que sa tension analogique soit comparée avec le signal d'entrée (qu'on souhaite convertir en numérique) et on arrête la conversion lorsque ces deux signaux analogiques sont proches. La dernière valeur numérique envoyé dans le CNA correspondra à la tension analogique d'entrée sur S\_IN. Le signal TEST du comparateur sera à 1 lorsque  $(E_+ - E_-) \geq 0$  sinon le signal TEST sera à 0. Ce signal TEST est câblé sur le bit 0 du port P4 du microcontrôleur.



On souhaite écrire une fonction CONV1, en langage C, permettant de convertir le signal analogique S\_IN en une valeur numérique. L'algorithme de conversion sera similaire à celui utilisé dans les convertisseurs analogiques/numériques à comptage.

- 4) Proposer la mise en œuvre de cette fonction CONV1 et indiquer le type de la valeur retournée. Indiquer l'ordre de grandeur du temps de conversion de cet algorithme.

**Exercice 5 :**

On dispose d'une carte microcontrôleur (type 80C592) disposant d'un bus I2C relié à plusieurs périphériques esclaves. Le microcontrôleur est le maître du bus. Les signaux SDA et SCL sont implantés sur les bits 7 et 6 du port P1 du microcontrôleur 80C592.

- 1) Précisez les conditions électriques sur les pattes 6 et 7 du port P1 pour pouvoir implanter un bus I2C.

Le chronogramme de la figure 1 (voir annexe 1) donne les signaux SCL et SDA du bus I2C dans le temps.

- 2) Décrire de manière textuelle (comme vu en cours) la trame I2C de la figure 1.  
 3) Indiquez les adresses des esclaves qui ont répondu ainsi que l'action sollicitée.

On souhaite, dans cette partie de l'exercice, transmettre vers un esclave de type EEPROM (16koctets) pour y écrire un bloc de 32 octets à partir de l'adresse 1000h dans sa mémoire. La procédure consiste à envoyer l'identifiant I2C (avec le bit W/R à 0) suivi de l'adresse (notée X) mémoire sur 2 octets pour précise la case mémoire à écrire dans l'EEPROM. On désignera par Xh et Xl, les 2 octets de l'adresse ( $X = Xh * 256 + Xl$ ).

On décide de réaliser cette opération en respectant l'algorithme suivant :

*Répéter 32 fois*

$S, (IDENT\_I2C, W), a, Xh, a, Xl, a, DATA_b, a, P;$

*Incrémenter X;*

*Fin répétition*

- 4) Indiquez le nombre total de bits nécessaires pour écrire les 32 octets utiles dans l'EEPROM.  
 5) Proposez une solution plus intéressante, en considérant que le registre d'adressage des octets de l'EEPROM est auto incrémenté à chaque écriture (ou lecture). Précisez le gain obtenu.

**Exercice 2 : Conception de circuits numériques**

On considère un mot A codé sur n bits, noté  $[a_{n-1} a_{n-2} \dots a_2 a_1 a_0]_b$ . On souhaite construire un opérateur itératif dont la sortie finale est un signal logique P tel que : P est à 1 si le nombre de bits à 1, dans le mot binaire A, est impair. P est à zéro dans le cas contraire.

- 1) Ecrire l'équation logique permettant d'obtenir le signal logique P.
- 2) Dédire une construction itérative, en considérant qu'à l'étape i (en parcourant i du poids faible au poids fort),  $P_i$  s'écrit sous la forme d'une fonction logique :  $P_i = F(P_{i-1}, a_i)$ .
- 3) Donner le schéma logique du bloc élémentaire.

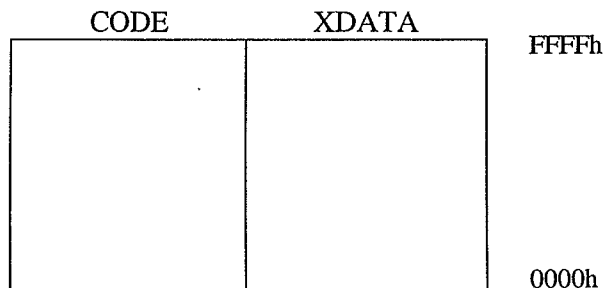
**Exercice 3 : Etude d'une carte microprocesseur**

Considérons le schéma logique incomplet d'une carte à microprocesseur (80C552) de l'annexe 1. En effet, certains signaux logiques ne sont pas câblés. Nous souhaitons compléter le câblage de cette carte tout en utilisant un circuit programmable CPLD étudié en cours. Il n'est pas demandé dans cet exercice d'utiliser le langage ABEL.

- 1) Indiquer les circuits qui ne peuvent pas fonctionner correctement et identifier le problème.

Nous disposons d'une documentation technique de cette carte microprocesseur, donnant les informations suivantes :

- ◆ Mémoire U2 accessible dans sa totalité dans l'espace XDATA à partir de l'adresse 0000H.
  - ◆ Mémoire U3 accessible dans les espaces XDATA et CODE à partir de l'adresse 8000H jusqu'à EFFFH.
  - ◆ L'espace mémoire F000h jusqu'à FFFFh n'est pas utilisé.
- 2) Préciser l'adresse de fin de l'espace mémoire occupé par le circuit U2.
  - 3) Préciser par un graphique simple (voir ci-dessous) l'organisation mémoire et l'espace (XDATA , CODE) occupés par les circuits U1, U2, U3. Identifier les espaces mémoire libres.

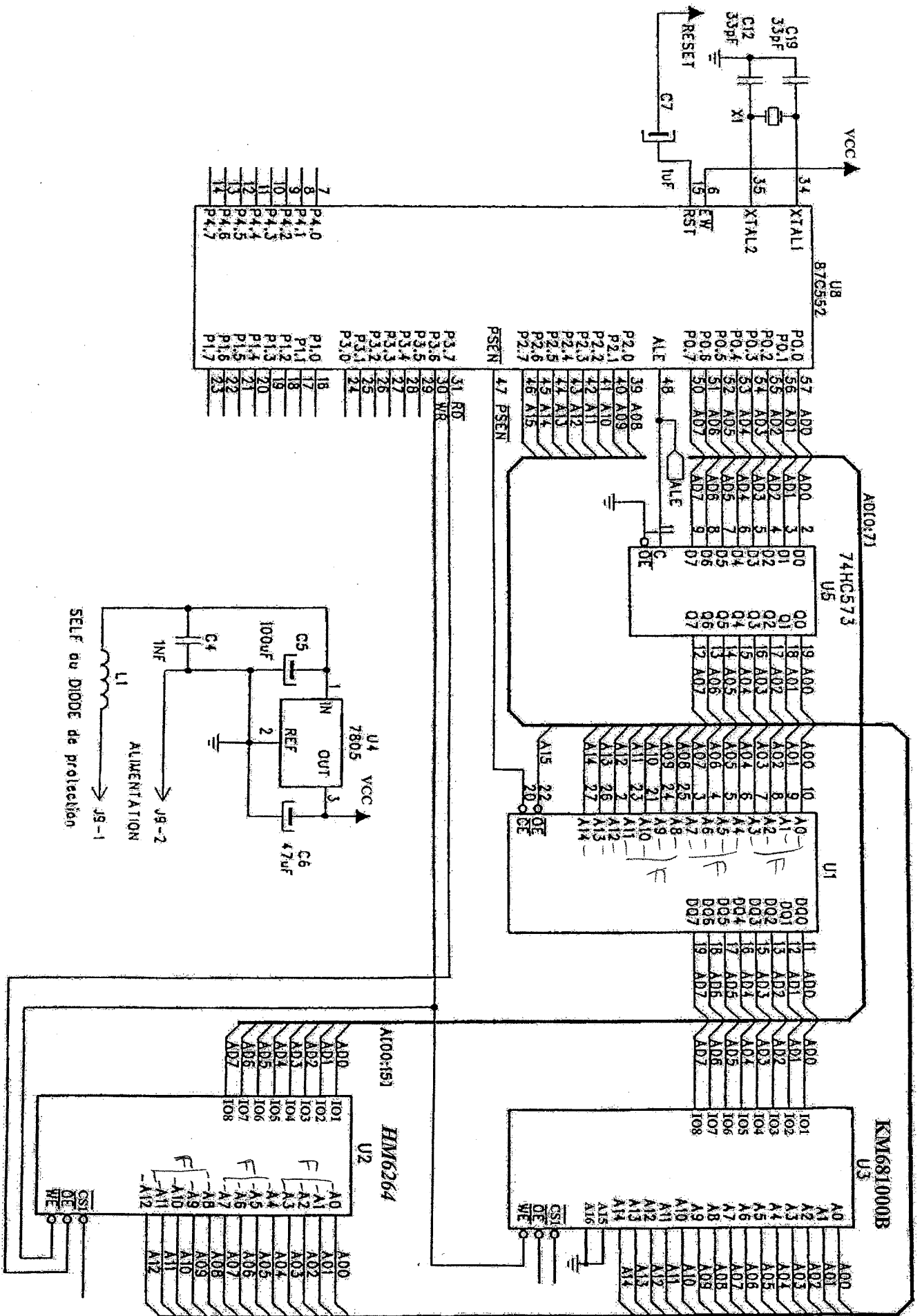


- 4) Donner les équations logiques des signaux manquants sur les circuits U2 et U3.
- 5) Indiquer, d'après le schéma de la carte microcontrôleur (et de la représentation du circuit U3), la taille mémoire réelle du circuit U3. Préciser l'espace mémoire non utilisé sur ce circuit.

On souhaite exploiter la mémoire libre du circuit U3 pour y implanter l'espace mémoire utilisé par le circuit U2. Cela a pour conséquence d'éliminer le circuit U2 qui devient inutile (car remplacé par une partie de l'espace mémoire de U3). On s'autorise alors d'étendre l'espace mémoire XDATA débutant à l'adresse 0000H grâce à la mémoire libre du circuit U3.

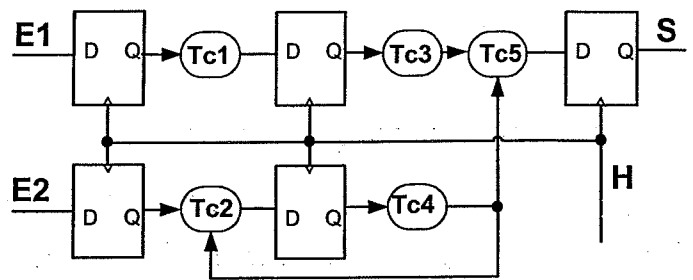
- 6) Indiquer par un graphique simple la nouvelle organisation mémoire de la carte microcontrôleur des espaces XDATA et CODE.
- 7) Dédire les nouvelles équations des signaux de contrôle du circuit U3 et préciser les modifications supplémentaires à apporter sur ce circuit.

**ANNEXE 1**



**Exercice 2 :**

Soit le schéma logique de la figure ci-contre:  
On supposera que les bascules ont un temps de SETUP, de HOLD et de PROPAGATION non nuls que l'on notera (respectivement)  $T_{su}=3ns$ ,  $T_h=2ns$  et  $T_{pd}=5ns$ .



1) Quelle est la fréquence maximale de l'horloge H permettant un bon fonctionnement du montage ? Justifiez votre réponse.

Temps de propagations des blocs combinatoires:  $14ns \leq T_{c1} \leq 17ns$ ,  $15ns \leq T_{c2} \leq 18ns$ ,  $10ns \leq T_{c3} \leq 12ns$ ,  $10ns \leq T_{c4} \leq 13ns$ ,  $4ns \leq T_{c5} \leq 7ns$ .

**Exercice 3 :**

On considère une carte microcontrôleur à base d'un Microcontrôleur 80C552. On souhaite adjoindre à la carte un circuit de Conversion Numérique->Analogique (CNA). Ce CNA permet de produire une tension analogique comprise entre 0.5 et 4.5 volts à partir d'une donnée numérique (non signée) sur 8 bits.

1) Donner la résolution en sortie du CNA (la plus petite variation de tension pouvant être générée).

Ce CNA dispose des signaux suivants :

- 8 bits de données D7 à D0.
- 1 bit  $\overline{CS}$  pour activer le boîtier.
- 1 signal de sortie analogique.
- VCC et GND, les deux signaux d'alimentation du circuit.

La carte microcontrôleur dispose d'un certain nombre de circuits annexes occupants le plan mémoire externe du microprocesseur. Ce plan d'organisation de la mémoire externe est le suivant :

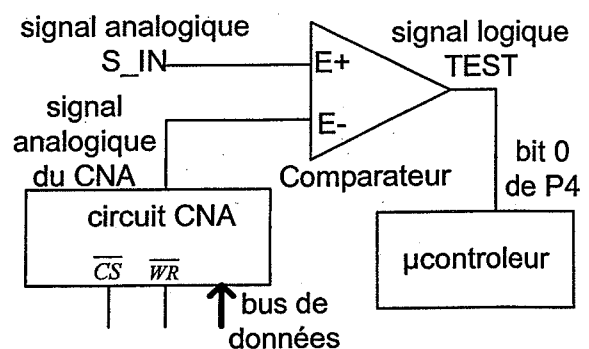
0000H à 7FFFH	Mémoire RAM externe RAM1
8000H à 9FFFH	Circuit annexe d'affichage vidéo
A000H à AFFFH	Zone libre
B000H à EFFFH	Circuit d'acquisition de données RDS
F000H à FFFFH	Zone libre

2) Proposer une zone mémoire libre pour implanter le circuit CNA. On considère que ce circuit pourra occuper au plus une page de 256 octets pour simplifier les équations de décodage d'adresses, dans ce cas quels sont les bits d'adresse intervenant dans le décodage?

3) Donner l'équation du bit de sélection du boîtier CNA et proposer un schéma d'implantation.

On souhaite étudier différents algorithmes de conversions analogique/numérique utilisant le CNA. On réalise pour cela le montage ci contre.

Le principe de la conversion analogique/numérique consiste à contrôler le circuit CNA de sorte que sa tension analogique soit comparée avec le signal d'entrée (qu'on souhaite convertir en numérique) et on arrête la conversion lorsque ces deux signaux analogiques sont proches. La dernière valeur numérique envoyé dans le CNA correspondra à la tension analogique d'entrée sur S\_IN. Le signal TEST du comparateur sera à 1 lorsque  $(E_+ - E_-) \geq 0$  sinon le signal TEST sera à 0. Ce signal TEST est câblé sur le bit 0 du port P4 du microcontrôleur.



On souhaite écrire une fonction CONV1, en langage C, permettant de convertir le signal analogique S\_IN en une valeur numérique. L'algorithme de conversion sera similaire à celui utilisé dans les convertisseurs analogiques/numériques à comptage.

- 4) Proposer la mise en œuvre de cette fonction CONV1 et indiquer le type de la valeur retournée. Indiquer de manière qualitative le temps de conversion de cet algorithme.

On souhaite maintenant mettre en œuvre un algorithme de conversion analogique/numérique plus performant. On s'oriente alors vers une technique dite par approximations successives utilisée dans certains convertisseurs CAN.

- 5) Proposer une mise en œuvre en C de cet algorithme de conversion sous forme d'une fonction CONV2 en précisant le type de la valeur retournée. Indiquer de manière qualitative le temps de conversion de cet algorithme.

On décide de mettre en œuvre un nouvel algorithme CONV3 basé sur l'algorithme à approximations successives modifié. La modification réside dans le fait que la valeur initiale à envoyer dans le CNA est la dernière valeur numérique convertie du signal analogique S\_IN. On suppose également qu'un programme fait appel à cette nouvelle fonction de conversion et qu'une valeur numérique du signal analogique est obtenue toutes les 100µs. Le signal analogique S\_IN a une variation de tension inférieure en valeur absolue à 0.15mV/µs dans le temps (SlewRate).

- 6) Soit  $X_n$  la valeur numérique convertie du signal S\_IN à l'instant n. Indiquer la plus grande et la plus petite variation de la valeur absolue de  $(X_n - X_{n-1})$ .

- 7) Proposez une mise en œuvre en C de la nouvelle fonction CONV3.

#### Exercice 4 :

On dispose d'une carte microcontrôleur (type 80C552) disposant d'un bus I2C relié à plusieurs périphériques esclaves. Le microcontrôleur est le maître du bus. Les signaux SDA et SCL sont implantés sur les bits 7 et 6 du port P1 du microcontrôleur 80C552.

- 1) Précisez les conditions électriques sur les pattes 6 et 7 du port P1 pour pouvoir implanter un bus I2C.

Le chronogramme de la figure 1 (voir annexe 1) donne les signaux SCL et SDA du bus I2C dans le temps.

- 2) Décrire de manière textuelle (comme vu en cours) la trame I2C de la figure 1.

- 3) Indiquez les adresses des esclaves qui ont répondu ainsi que l'action sollicitée.

On souhaite transmettre vers un esclave de type EEPROM pour y écrire un bloc de 16 octets à partir de l'adresse 10h. La procédure consiste à écrire un octet par trame I2C à l'adresse X ( $10h \leq X \leq 1Fh$ ) de l'EEPROM. On répétera pour chaque octet à écrire la trame suivante : S, (ADR,W), a, X, a, DATA<sub>1</sub>, a, P.

- 4) Précisez ce qui est envoyé par le maître et ce que répond l'esclave dans la trame I2C.
- 5) Indiquez le nombre total de bits nécessaire pour écrire les 16 octets utiles dans l'EEPROM.
- 6) Proposez une solution plus intéressante, en considérant que le registre d'adressage des octets de l'EEPROM est auto incrémenté à chaque écriture (ou lecture). Précisez le gain obtenu.

**EXAMEN PARTIEL UE441**

Documents autorisés : documents distribués en cours, TD, TP. Tout autre document est interdit.

Questions liées au cours

1) Considérons l'équation logique suivante :  $S = A + B$  avec  $S.OE = A$

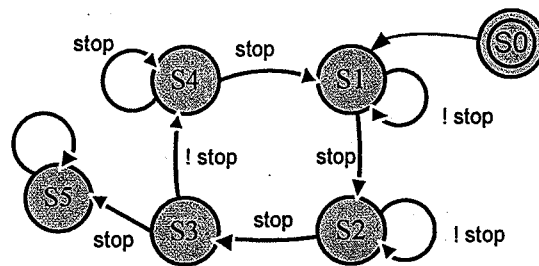
- ◆ Préciser le comportement de la sortie en fonction de A et B.

On ajoute une résistance R (10Kohm) entre la sortie S et le +5V (résistance de PULL-UP).

- ◆ Cela modifie-t-il le comportement de la sorte S. Déduire dans ce cas la nouvelle équation logique réalisée par cette sortie S.

2) Considérons le graphe de l'automate suivant:

- ◆ Précisez si cet automate est déterministe.



3) considérons la mémoire DATA internes d'un microcontrôleur 80C552, organisée en BIG ENDIAN :

Adresse mémoire :	i	i+1	i+2	i+3	i+4	i+5	i+6	i+7	i+8
Contenu	82h	E3h	42h	06h	CAh	0Bh	0Bh	10h	30h

- ◆ Indiquez la valeur d'une variable X entière (int) codée sur 16 bits et implantée à partir de l'adresse i+4.

Exercice 1 :

On souhaite réaliser un compteur perpétuel (modulo 4) capable d'effectuer un comptage ou décomptage sur 2 bits [C1,C0] en binaire non signé. Un signal d'entrée E permet de choisir à tout moment le sens de comptage. En effet, pour toute valeur binaire actuelle du compteur, la valeur future sera fonction d'un signal de commande E. Si E=0 cette valeur future sera le résultat d'un comptage (+1). Dans le cas où E=1, la valeur future sera le résultat d'un décomptage (-1).

On se propose alors de réaliser un automate dont la variable d'état sera la valeur de sortie de notre compteur (C1,C0). Le signal d'horloge de cet automate est H.

- 1) Dessinez le graphe de l'automate du cahier des charges ci-dessus. Déduire la table d'états-transitions.
- 2) Déduire le signal E à entrer pour obtenir la séquence [C1,C0] suivante: 00 01 10 01 10 11 00 01 00
- 3) Donnez les équations logiques de D1 et D0 (les entrées des bascules D) en fonction de E, C1, C0.
- 4) A partir d'une portion d'un PAL (donné en figure 1 et **à rendre avec votre copie**), implanter ce compteur (*Marquez d'une croix sur la matrice des ET les fusibles à maintenir*). Le signal E est fixé sur l'entrée 2 (voir figure 1). *Utilisez obligatoirement les rebouclages internes des sorties et ne pas considérer le contrôle 3 états.*
- 5) Calculer la fréquence maximum de fonctionnement de ce compteur en supposant que E reste fixe.

Cette Partie (ci dessous) est à rendre avec votre copie

- 6) Décrire en langage ABEL cet automate. On ne tiendra pas compte des numéros de broches associés aux différents signaux.
- 7) A partir d'une portion d'un PAL (donné en figure 1 et à rendre avec votre copie), implanter ce système séquentiel (Marquez d'une croix sur la matrice des ET les fusibles à maintenir). Le signal E est fixé sur l'entrée 2 (voir figure 1). Utilisez obligatoirement les rebouclages internes des sorties et ne pas considérer le contrôle 3 états.
- 8) Calculer de nouveau la fréquence maximum de fonctionnement de ce système implanté dans le PAL en supposant que E reste fixe.

**Cette Partie est à rendre avec votre copie**

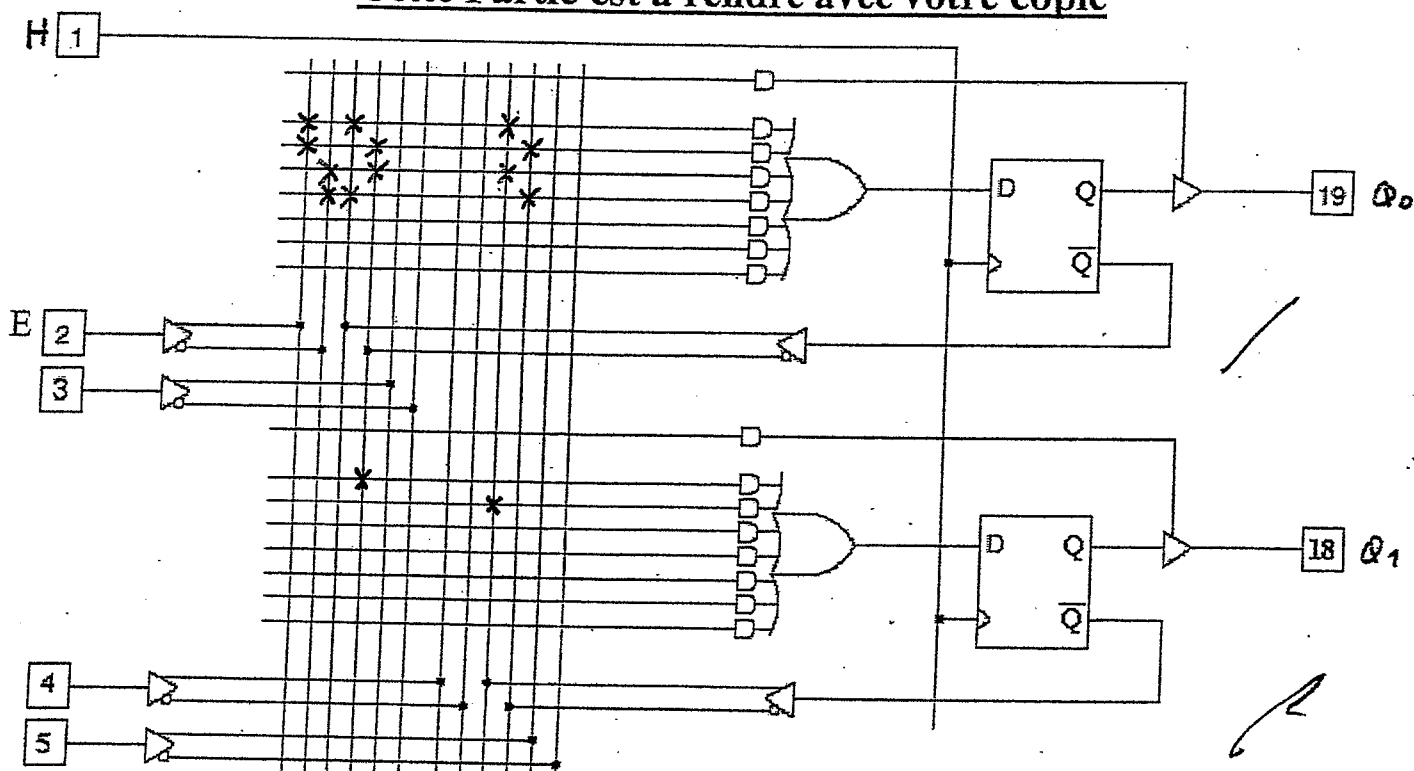


Figure 1 : partie d'un PAL.

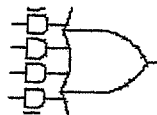
Temps de propagations :

Pour la bascule D :  $T_{pd} = 3\text{ ns}$

$T_{su} = 2\text{ ns}$

$T_h = 1\text{ ns}$

Pour les parties combinatoires :



Sortie =

5 ns

2 ns

1 ns

**Exercice 2 :**

Considérons le schéma logique d'une carte à microprocesseur (80C552) de l'annexe 1. Nous souhaitons remplacer tous les circuits combinatoires et la fonctionnalité du latch (circuit U5) par un circuit programmable de type ispLSI1016. Les circuits concernés sont les portes logiques discrètes (U6A, U6B, U6D, U7A et U7B) ainsi que le circuit latch D 74HC573 (U5).

Une documentation technique est donnée pour le circuit 74HC573. Le circuit CPLD (ispLSI1016) utilise une horloge de fréquence  $F = 30\text{ Mhz}$ .

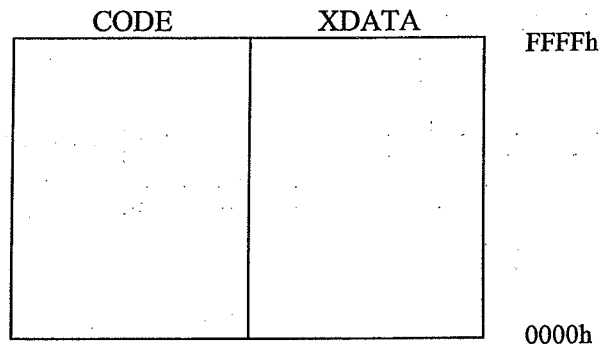
Le circuit latch D, 74HC573 sera spécifié en ABEL comme étant un registre (8 bits) synchrone doté d'un signal enable (autorisation d'écriture). Vous trouverez des fiches techniques pour cet exercice en Annexe 2. Le signal OE du latch U5 étant toujours à 0, les sorties du latch seront toujours actives.



### Exercice 3 : Etude d'une carte microprocesseur

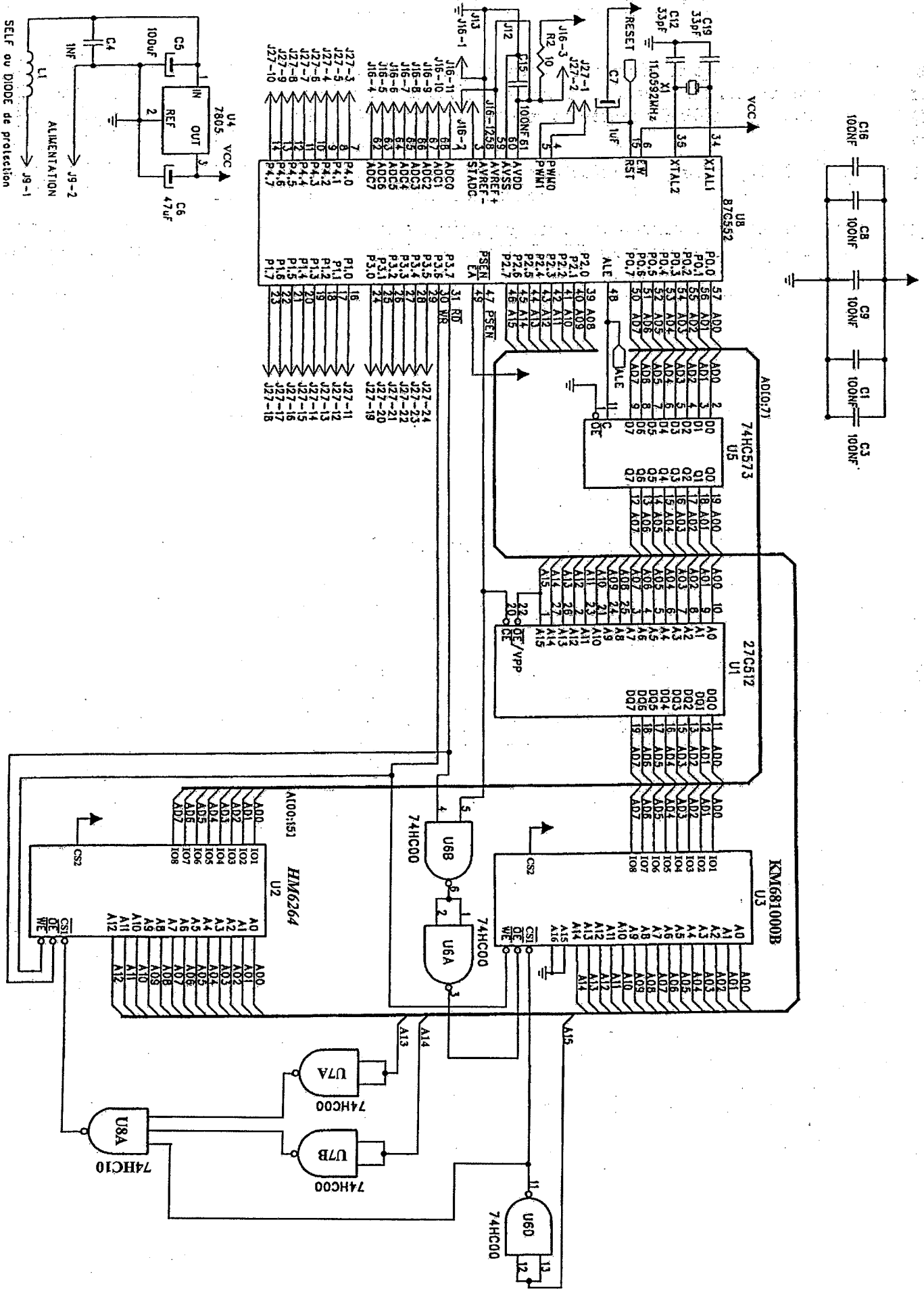
Considérons le schéma logique d'une carte à microprocesseur (80C552) de l'annexe 1. Nous souhaitons remplacer tous les circuits combinatoires par un circuit programmable de type ispLSI1016. Le circuit CPLD (ispLSI1016) utilise une horloge de fréquence  $F=30\text{Mhz}$ . (Annexe 2 donne certaines documentations des circuits utilisés).

- 1) Indiquer, pour chaque mémoire de la carte, sa nature (ROM ou RAM) sa taille totale (en Koctets) ainsi que la taille réellement utilisée.
- 2) Préciser par un graphique simple (voir ci-dessous) l'organisation mémoire et l'espace (XDATA , CODE) occupés par les circuits U1, U2, U3. Indiquer les espaces mémoire libres.



- 3) Déduire les équations logiques de décodage de ces circuits mémoires.
- 4) Indiquez les zones d'adressages mémoires libres (n'adressant aucun composant).
- 5) Préciser les signaux de contrôle à utiliser par le circuit CPLD pour réaliser les fonctions de décodages d'adresses en remplacement des portes logiques.
- 6) Nous constatons que la capacité mémoire du circuit U3 (KM681000B) n'est pas totalement exploitée. Nous décidons alors d'exploiter cette capacité inutilisée pour remplacer le circuit U2. Il est important de respecter le type de l'espace mémoire occupé par U2 et par U3. Proposez une solution pour cette mise en œuvre en précisant les nouvelles équations de décodage du circuit U3 (équation logique de CS et OE actif au niveau logique bas).

# ANNEXE 1

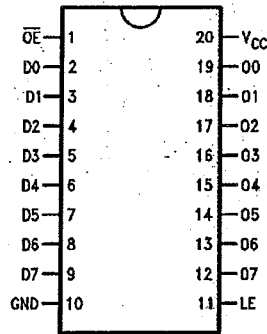


# ANNEXE 2

## DOCUMENTATION SIMPLIFIEE DU CIRCUIT SN74ALS373/573

Si LE=1 alors le latch est transparent (laisse passer les données).

Si LE=0 alors la dernière valeur Di est mémorisée.



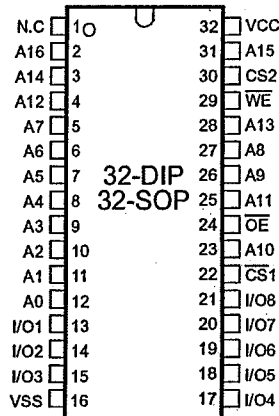
### Function Tables

$\overline{OE}$	LE	D	Output O
L	H	H	H
L	H	L	L
L	L	X	Q <sub>O</sub>
H	X	X	Z

L = LOW State  
H = HIGH State  
X = Don't Care  
Z = High Impedance State  
Q<sub>O</sub> = Previous Condition of O

## DOCUMENTATION SIMPLIFIEE DE LA MEMOIRE RAM : KM681000B

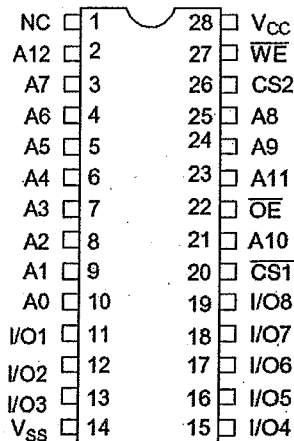
Le signal CS2 sera forcé à 1 et seul CS1 sera utilisé comme chipselect (CS1 actif à 0)



Name	Function
$\overline{CS1}, \overline{CS2}$	Chip Select Inputs
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
A0~A16	Address Inputs
I/O1~I/O8	Data Inputs/Outputs
Vcc	Power
Vss	Ground
N.C	No Connection

## DOCUMENTATION SIMPLIFIEE LA MEMOIRE RAM : HM6264

Le signal CS2 sera forcé à 1 et seul CS1 sera utilisé comme chipselect (CS1 actif à 0)



Name	Function
$\overline{CS1}, \overline{CS2}$	Chip Select Inputs
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
A0~A12	Address Inputs
I/O1~I/O8	Data Inputs/Outputs
Vcc	Power
Vss	Ground
N.C	No Connection

### Circuits logiques utilisés:

74HC00 : NAND à 2 entrées.

74HC10 : NAND à 3 entrées.

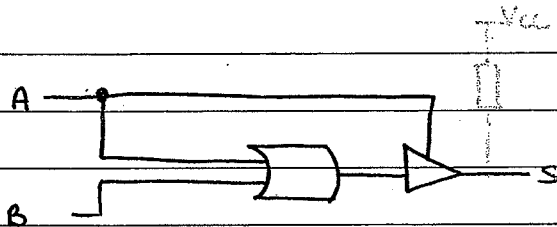
Partiel 444 Nov 2007

Fait par un élève

Questions de cours

1.

\*



sans résistance:

A	B	S
0	0	HZ
0	1	HZ
1	0	1
1	1	1

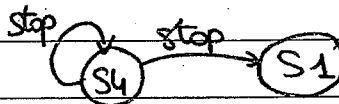
HZ: haute impédance

\* Si on ajoute la résistance (en vert):

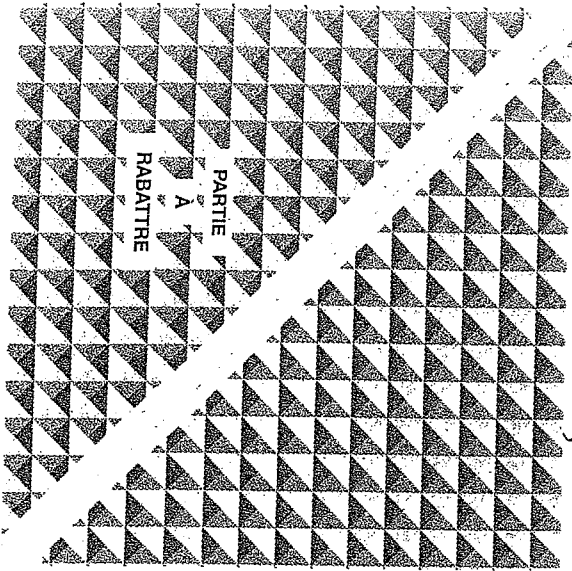
A	B	S
0	0	1
0	1	1
1	0	1
1	1	1

$S = 1$

2. On remarque que de l'état S4, deux états partent avec la même transition "stop":



donc si on a S4 et stop, on ne connaît pas l'état suivant, donc l'automate n'est pas déterministe



3. La variable codée sur 16 bits à partir de  $i+4$  s'écrit en hexadécimal:

$$CA\phi B_f$$

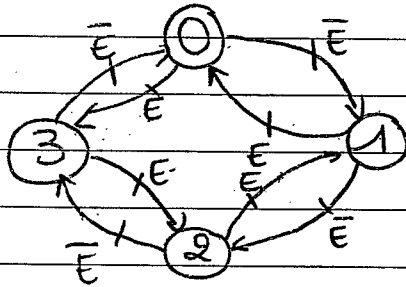
(big endian, donc les bits de poids fort sont écrits en premier dans la mémoire).

$$\text{donc } X = 12 \times 16^3 + 10 \times 16^2 + 0 + 11$$

$$X = 51723$$

### Exercice 1.

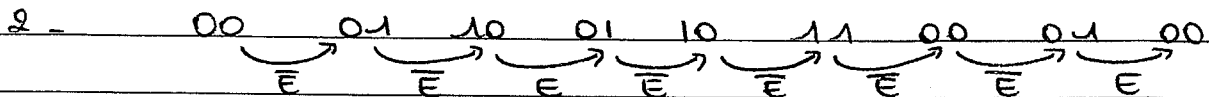
1



Les fronts d'horloge sont simplifiés.

1

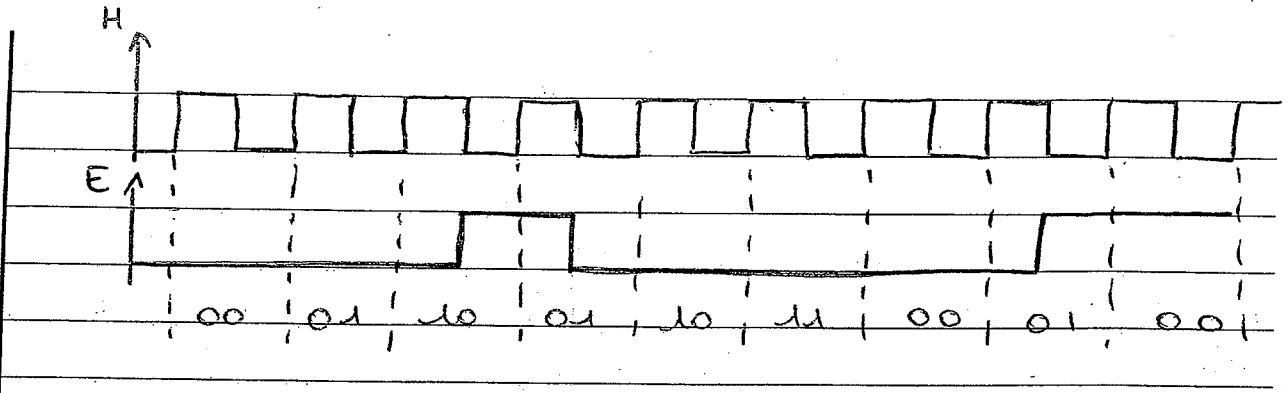
E	État $n$		État $n+1$	
	$C_1$	$C_0$	$C_1$	$C_0$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	1
1	1	1	1	0



Notez avec exactitude votre numéro de table ou d'inscription.

Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

05



B <sub>-</sub>	D <sub>1</sub>	C <sub>1</sub>	0	0	1	1	D <sub>0</sub>	C <sub>0</sub>	0	0	1	1
	E	0	0	1	0	1	E	1	0	0	1	1
	1	1	0	1	0	0	1	1	0	0	1	1

1

$$D_1 = E (\bar{C}_0 \bar{C}_1 + C_0 C_1) + \bar{E} (C_0 \bar{C}_1 + \bar{C}_0 C_1); \quad D_0 = \bar{C}_0$$

$$D_1 = E (C_0 \oplus C_1) + \bar{E} (C_0 \oplus C_1)$$

donc

$$D_0 = \bar{C}_0$$

$$D_1 = E \oplus (C_0 \oplus C_1)$$

④

05

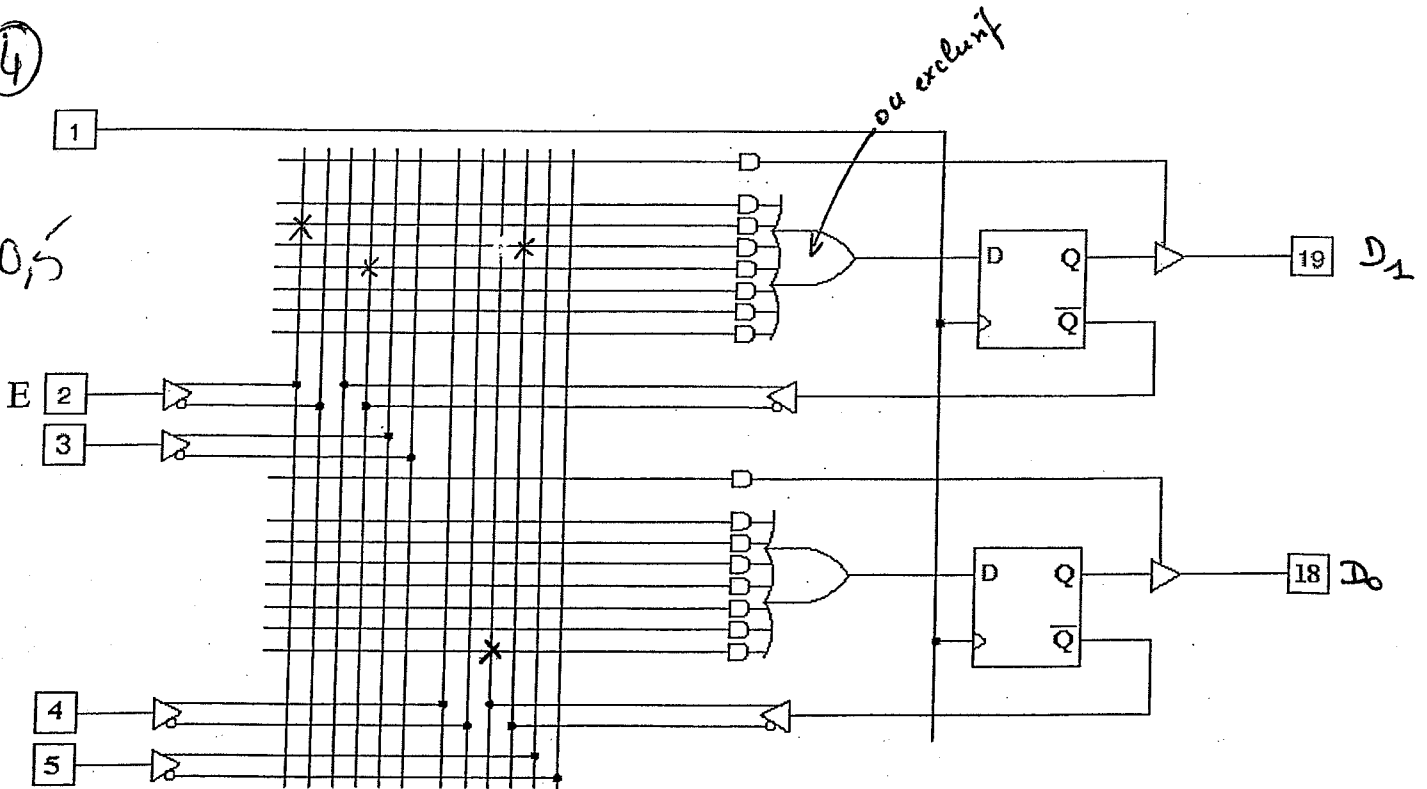


Figure 1 : partie d'un PAL.

Notez avec exactitude votre numéro de table ou d'inscription.

Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

$$5 - T_{\min} = T_{\text{pd}} + T_{\text{combi}} + T_{\text{su}}$$

$$T_{\text{combi}} = T_{\text{combi max}} = 5 + 2 = 7 \text{ ns}$$

donc  $T_{\min} = 12 \text{ ns}$

$$f_{\max} = 83,3 \text{ MHz}$$

### Exercice 2

$a_3$	$a_2$	$a_1$	$a_0$	$X_3$
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$X_3$	$a_3$	$a_2$	$a_1$	$a_0$
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

$$X_3 = a_3 a_0 + a_3 a_2 + a_1 a_2 + a_0 a_2 + a_3 a_1 + a_3 a_0$$

Notez avec exactitude votre numéro de table ou d'inscription.

Il est interdit au candidat de signer sa copie ou d'y mettre un signe quelconque pouvant identifier l'auteur de la copie.

$$X_3 = c_0(c_1 + c_2 + c_3) + c_1(c_2 + c_3) + c_2 c_3$$

1

2. 
$$Y_i = Y_{i-1} + a_i$$

3.

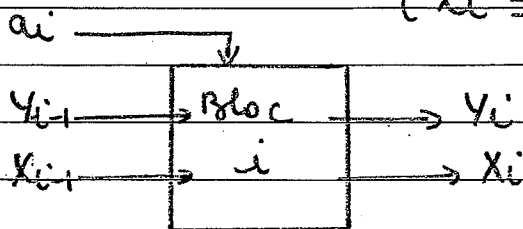
$X_{i-1}$	$Y_{i-1}$	$a_i$	$X_i$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	1	0	1
1	1	1	1

(Rmq:  $X_{i-1} Y_{i-1}$  est impossible)

1,5

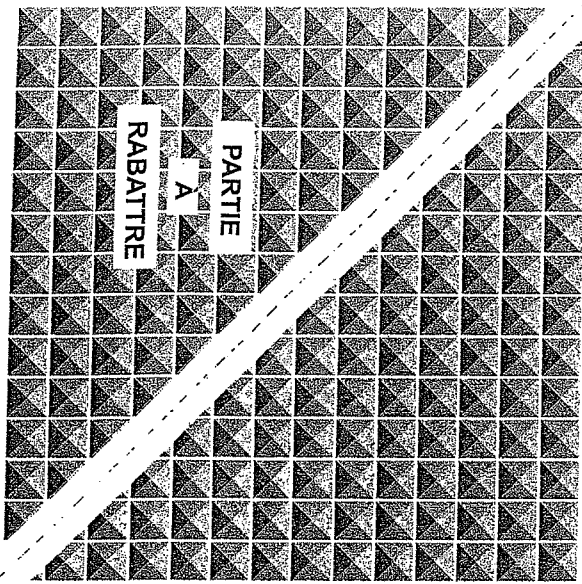
donc 
$$X_i = X_{i-1} + Y_{i-1} a_i$$

4. In fine, nous avons: 
$$\begin{cases} Y_i = Y_{i-1} + a_i \\ X_i = X_{i-1} + Y_{i-1} a_i \end{cases}$$

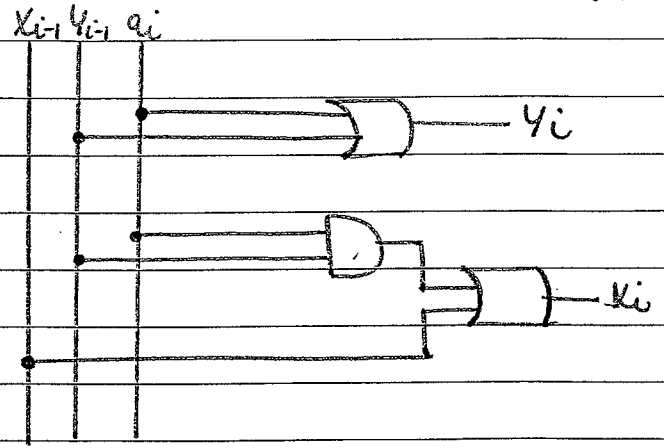


1

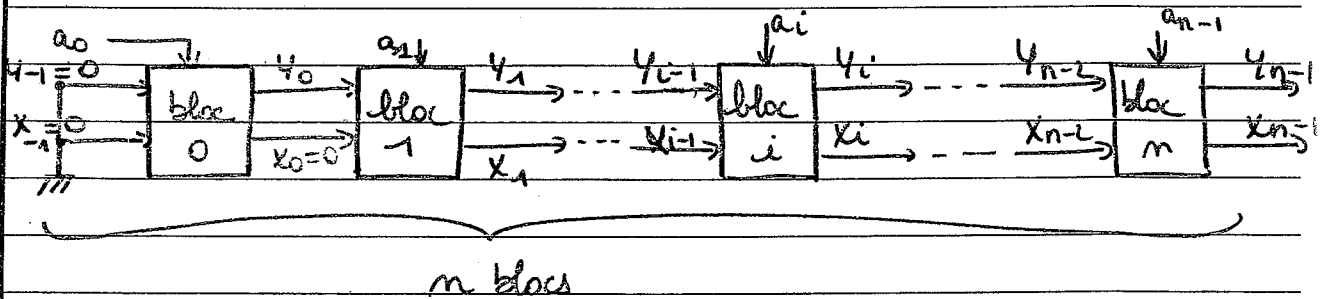




Avec l'architecture (schéma logique):



La construction itérative est de la forme:



Temps de propagation total (= retard entre la présentation de A en entrée et la stabilité de la sortie):

$$T_{\text{total}} = nT$$

6. Avantage d'une telle construction:

- on ne conçoit qu'un type de bloc qui permet d'effectuer des calculs quelque soit  $n$  (alors qu'à la que 1, l'équation ne fonctionne que pour  $n=3$ ) en cascade les blocs;
- quand  $n$  augmente, on utilise moins de portes logiques et la conception itérative, donc on gagne en rapidité.

### Exercice 3

1. \* Sur la carte, il y a 3 mémoires:

-  $U_2$  est une RAM, comme indiqué en Annexe 9) (ce qu'on peut voir puisqu'elle possède une entrée WE, donc on peut écrire dedans);

- de même  $U_3$  est une RAM;

- en revanche  $U_1$  ne possède pas d'entrée WE: c'est une ROM.

\* On compte le nombre de pattes d'adresse pour connaître la taille des mémoires:

$U_1$ : ROM 16 pattes  $\rightarrow$  64 ko (32 ko utilisés car  $\overline{OE} = A_{15}$ )

$U_2$ : RAM 13 pattes  $\rightarrow$  8 ko (tous utilisés)

$U_3$ : RAM 17 pattes  $\rightarrow$  128 ko  $\rightarrow$  15 pattes utilisées

$\hookrightarrow$  32 ko utilisés

On voit que pour  $U_3$  toute la mémoire n'est pas adressable car deux pattes sont reliées à la masse ( $A_{16}$  et  $A_{15}$ )

2 \* Il n'y a qu'une seule ROM, donc elle est dans l'espace CODE (puisque au démarrage la RAM est vierge et il faut que le microcontrôleur ait un programme à lire).

donc  $U_1$  est dans CODE, et comme  $\overline{OE} = A_{15}$ ,  $U_1$  contient les adresses de  $0000_h$  à  $7FFF_h$ .

\* Le câblage du "chip select" de  $U_2$  implique que le microcontrôleur lit et écrit dans  $U_2$  les adresses (16 bits) du type  $0000_{12} a_{11} a_{10} a_9 a_8 a_7 a_6 a_5 a_4 a_3 a_2 a_1 a_0$  (adresses situées entre  $0000_h$  et  $1FFF_h$ ).

\* Le câblage du "chip select" de  $U_3$ , lui, indique que le microcontrôleur lit et écrit dans  $U_3$  les adresses du type:

$1 a_{14} a_{13} a_{12} \dots a_3 a_2 a_1 a_0$   
(adresses situées entre  $8000_h$  et  $FFFF_h$ ).

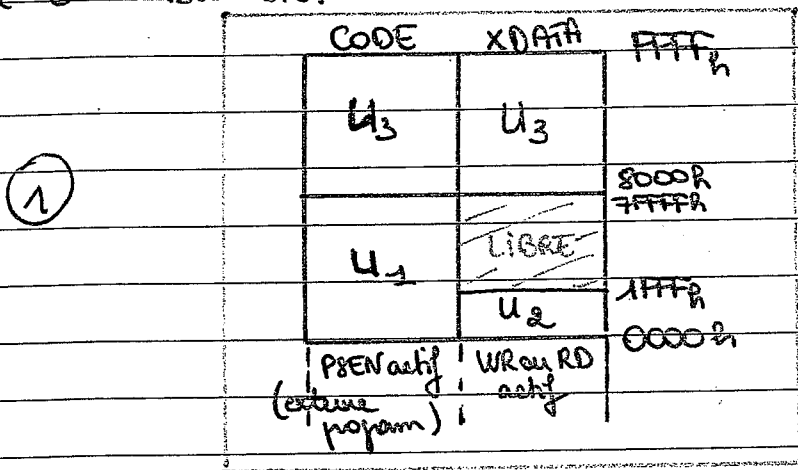
\* Il reste à déterminer si  $U_2$  et  $U_3$  sont dans code ou data.  
Pour  $U_2$ ,  $\overline{WR} = \overline{WE}$   
 $\overline{RD} = \overline{OE}$  } donc  $U_2$  est dans l'espace externe données (DATA).

165

Pour U3,  $\overline{WE} = \overline{WR}$   
 $\overline{OE} = \overline{\overline{PSEN} \cdot RD}$   
 $OE = PSEN + RD$

Autrement dit, U3 est à la fois dans l'espace DATA et l'espace CODE pour les adresses  $8000_h$  à  $FFFF_h$ ; en effet, le PSEN indique que le microcontrôleur lit du CODE (Program Store Enable) alors que le RD indique la lecture de données (DATA)

\* En conclusion:



3- U1 :  $\overline{OE} = A_{15}$   
 $\overline{CE} = \overline{PSEN}$

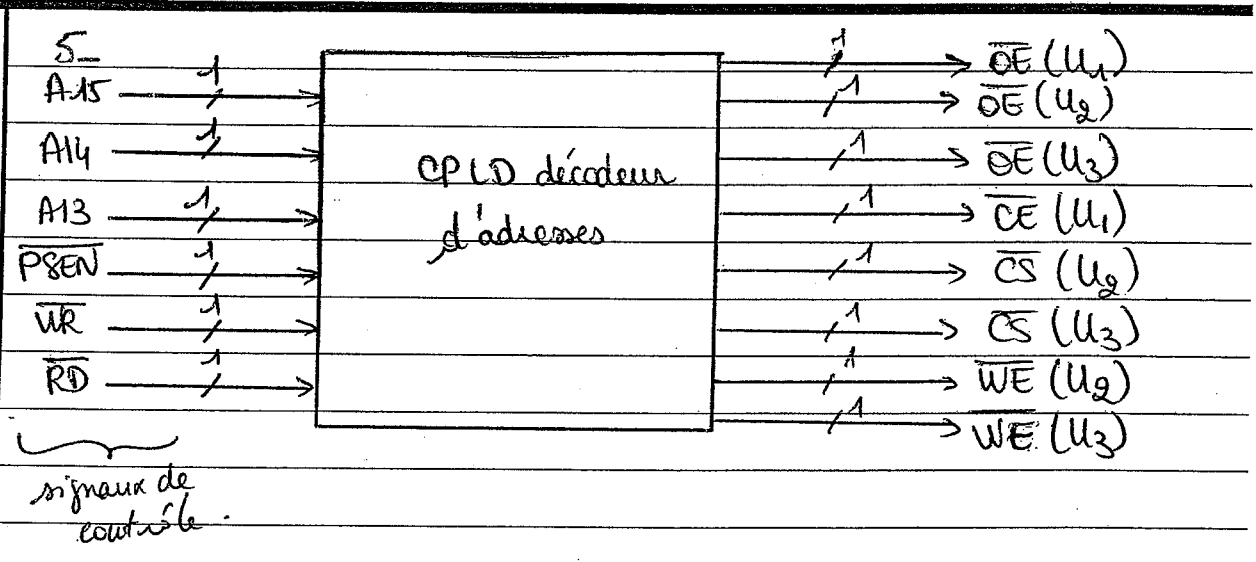
①, 1, 5

U2 :  $\overline{OE} = \overline{RD}$   
 $\overline{WE} = \overline{WR}$   
 $\overline{CS} = \overline{A_{13} \cdot A_{14} \cdot A_{15}}$  (ou  $\overline{CS} = A_{13} + A_{14} + A_{15}$  : le clup est actif si  $A_{13} = 0$  et  $A_{14} = 0$  et  $A_{15} = 0$ )

U3 :  $\overline{OE} = \overline{\overline{PSEN} \cdot RD} = PSEN + RD$   
 $\overline{WE} = \overline{WR}$   
 $\overline{CS} = \overline{A_{15}}$

4- Toute adresse comprise entre  $1FFF_h$  et  $7FFF_h$  n'adresse aucun composant si  $\overline{PSEN} = 0$  (zone libre dans l'espace XDATA).  
 Dès que  $\overline{PSEN} = 1$ , toutes les adresses adressent un composant (dans l'espace CODE).

015

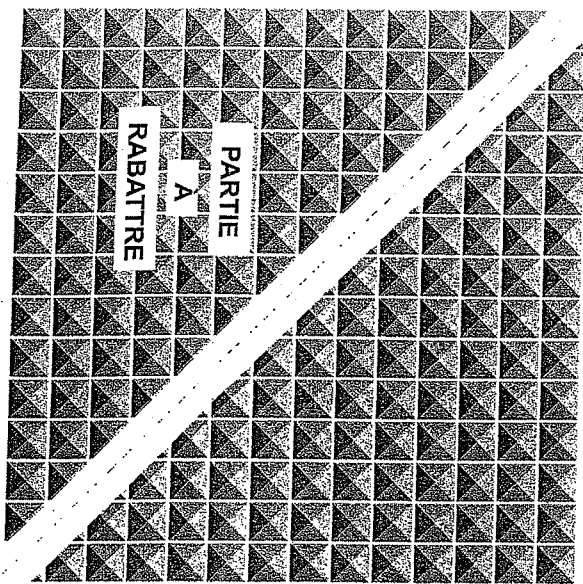


6. Récapitulons:

$U_2$  est une RAM de 8ko de l'espace XDATA

$U_3$  est une RAM de 128ko dont 32ko sont utilisés dans les deux espaces XDATA et CODE

↳ on veut remplacer  $U_2$  par la partie de  $U_3$  non utilisée. Les adresses adressées par le microcontrôleur à l'espace données correspondant à  $U_2$  sont :  $0000_n$  à  $1FFF_n$ . Les emplacements mémoire correspondant dans  $U_3$  sont <sup>en partie</sup> utilisés, parce que  $A_{15}$  et  $A_{16}$  sont à la masse, du coup les adresses  $8000_n$  à  $FFFF_n$  (des  $\mu C$ ) sont physiquement stockées dans  $0000_n$  à  $07FFF_n$  de  $U_3$ .



\* La première chose à faire est donc de relier A15 (du port de sortie du µC) à la patte A15 de U3 → l'espace situé entre 0000<sub>h</sub> et 7FFF<sub>h</sub> devient libre.

\* U2 était utilisé en XDATA, <sup>uniquement</sup> il ne faut donc pas que les adresses 0000<sub>h</sub> à 1FFF<sub>h</sub> soient à la fois en XDATA et en CODE comme les adresses 8000<sub>h</sub> à FFFF<sub>h</sub> (il y aurait conflit avec la ROTU1!).

donc on écrit:

$\overline{OE} = \overline{PSEN} \cdot A15 + RD$  : si l'adresse est située entre 8000<sub>h</sub> et FFFF<sub>h</sub>, A15 = 1 et on continue d'utiliser ces adresses en XDATA et en CODE. En revanche si A15 = 0, on n'utilise les adresses 0000<sub>h</sub> à 1FFF<sub>h</sub> qu'en XDATA.

On a toujours  $\overline{WE} = \overline{WR}$

On n'a plus  $\overline{CS} = \overline{A15}$

mais  $\overline{CS} = \overline{A15 + A15 \cdot \overline{A14} \cdot \overline{A13}}$

(U3 actif lorsqu'on s'adresse à une zone mémoire entre 0000<sub>h</sub> et 1FFF<sub>h</sub>).

\* En conclusion,

- on relie A15 du µC à la patte A15

- on change de décodage d'adresses:

$$\begin{cases} \overline{OE} = \overline{PSEN} \cdot A15 + RD \\ \overline{WE} = \overline{WR} \\ \overline{CS} = \overline{A15 + A15 \cdot \overline{A14} \cdot \overline{A13}} \end{cases}$$



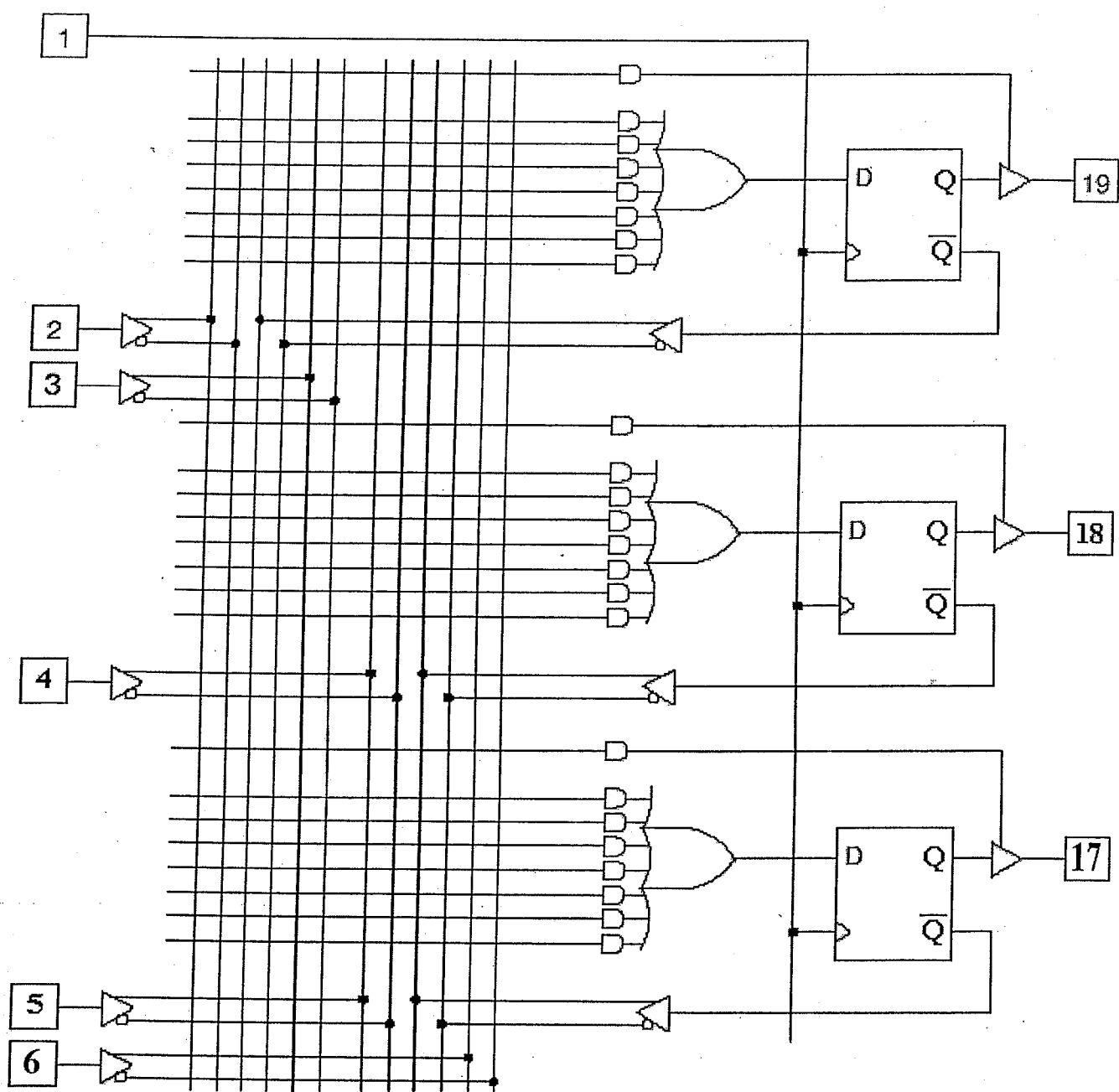


Figure 1 : partie d'un PAL.

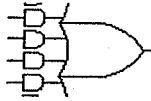
Temps de propagations :

Pour la bascule D :  $T_{pd} = 3\text{ns}$

$T_{su} = 2\text{ns}$

$T_h = 1\text{ns}$

Pour les parties combinatoires :



5 ns



2 ns



1 ns

Temps de propagation =

**Cette feuille est à rendre avec votre copie**

NOM:.....

PRENOM : .....

9

**Exercice 3 :**

Considérons le schéma logique incomplet d'une carte à microprocesseur (80C552) de l'annexe 1. En effet, certains signaux logiques ne sont pas câblés. Nous souhaitons compléter le câblage de cette carte tout en utilisant un circuit programmable de type ispLSI1016. Il n'est pas demandé dans cet exercice d'utiliser le langage ABEL.

1) Indiquer les circuits qui ne peuvent pas fonctionner correctement et identifier le problème.

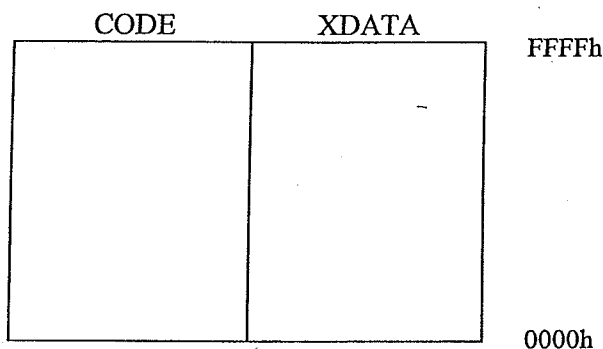
Nous disposons d'une documentation technique de cette carte microprocesseur, donnant les informations suivantes :

◆ Mémoire U2 accessible dans sa totalité dans l'espace XDATA à partir de l'adresse 0000H.

◆ Mémoire U3 accessible dans l'espace XDATA à partir de l'adresse 8000H jusqu'à FFFFH.

2) Préciser l'adresse de fin de l'espace mémoire occupé par le circuit U2.

3) Préciser par un graphique simple (voir ci-dessous) l'organisation mémoire et l'espace (XDATA , CODE) occupés par les circuits U1, U2, U3. Indiquer les espaces mémoire libres.



4) Donner les équations logiques des signaux manquants sur ces circuits U2 et U3.

5) Indiquer, d'après le schéma de la carte microcontrôleur (et de la représentation du circuit U3), la taille mémoire réelle du circuit U3. Préciser l'espace mémoire libre non utilisé sur ce circuit.

On souhaite exploiter la mémoire libre du circuit U3 pour y implanter l'espace mémoire utilisé par le circuit U2. Cela a pour conséquence d'éliminer le circuit U2 qui devient inutile (car remplacé par une partie de l'espace mémoire de U3). On s'autorise alors d'étendre l'espace mémoire débutant à l'adresse 0000H grâce à la mémoire libre du circuit U3.

6) Indiquer par un graphique simple la nouvelle organisation mémoire de la carte microcontrôleur des espaces XDATA et CODE.

7) Déduire les nouvelles équations des signaux de contrôle du circuit U3.



# ANNEXE 1

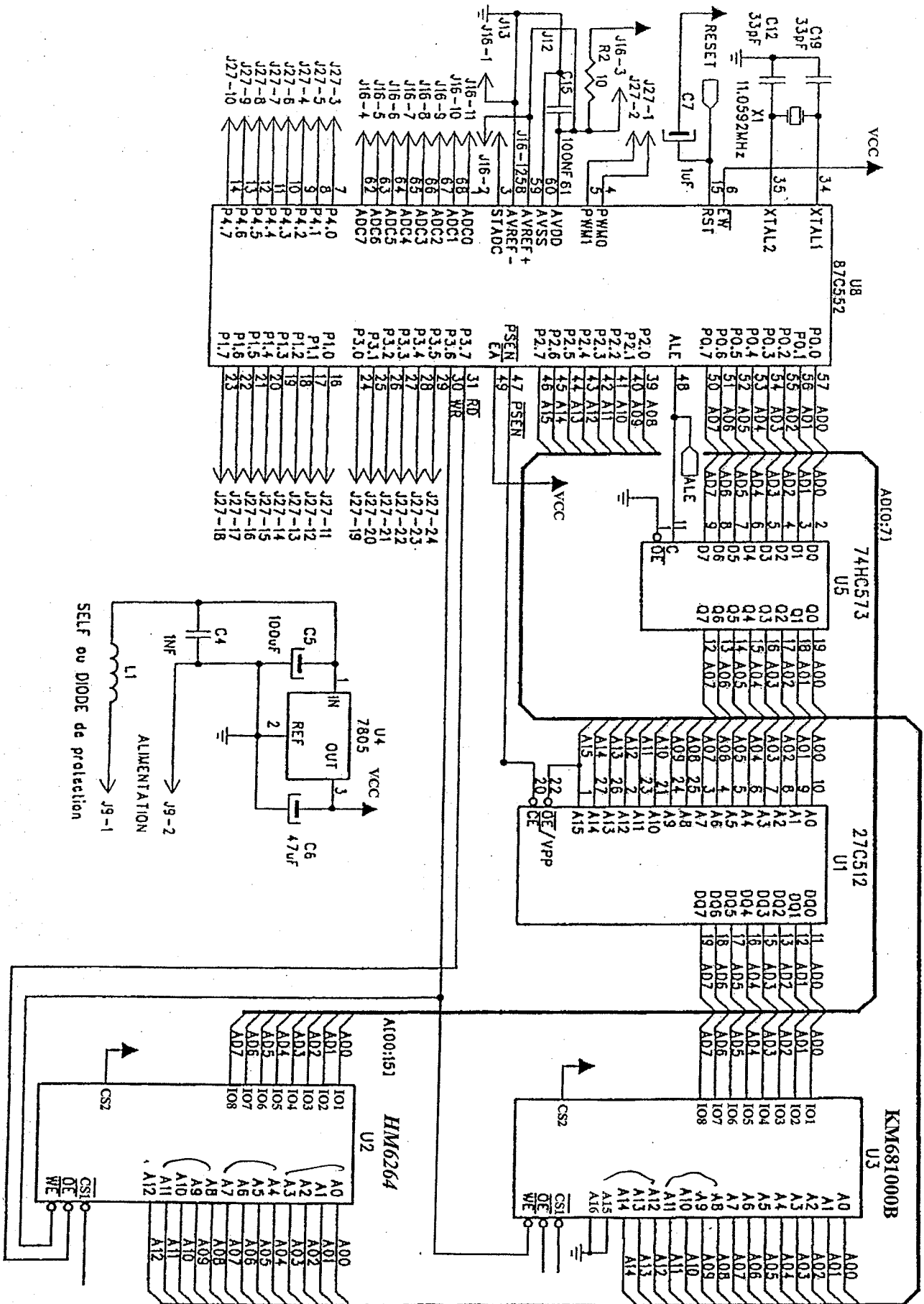


Schéma logique de la carte microcontrôle pour l'exercice 3

# ANNEXE 2

## DOCUMENTATION SIMPLIFIEE DU CIRCUIT SN74ALS373/573

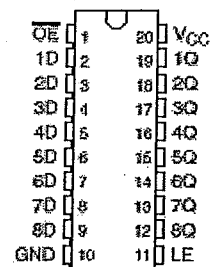
SN54ALS573C, SN54AS573A ... J OR W PACKAGE  
SN74ALS573C, SN74AS573A ... DW OR N PACKAGE  
(TOP VIEW)

Si LE=1 alors le latch est transparent  
(laisse passer les données).

Si LE=0 alors la dernière valeur Di est  
méorisée.

FUNCTION TABLE  
(each latch)

INPUTS			OUTPUT
OE	LE	D	Q
L	H	H	H
L	H	L	L
L	L	X	Qp
H	X	X	Z

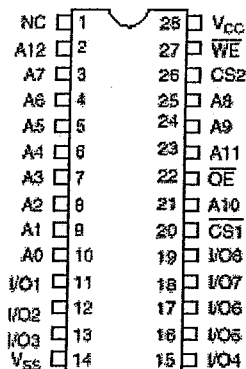


## DOCUMENTATION SIMPLIFIEE DE LA MEMOIRE RAM : HM6264

HM6264BLP/BLSP/BLFP Series

### Pin Description

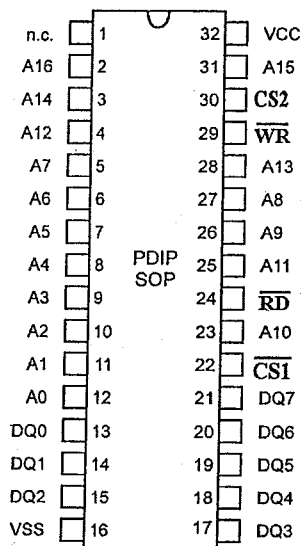
Le signal CS2 sera forcé à 1 et seul CS1 sera  
utilisé comme chipselect (CS1 actif à 0)



Pin name	Function
A0 to A12	Address input
I/O1 to I/O8	Data input/output
CS1	Chip select 1
CS2	Chip select 2
WE	Write enable
OE	Output enable
NC	No connection
V <sub>cc</sub>	Power supply
V <sub>ss</sub>	Ground

## DOCUMENTATION SIMPLIFIEE DE LA MEMOIRE RAM : KM681000B

Le signal CS2 sera forcé à 1 et seul CS1 sera  
utilisé comme chipselect (CS1 actif à 0)



Signal Name	Signal Description
A0 - A16	Address Inputs
DQ0 - DQ7	Data In/Out
CS1	Chip Enable 1
CS2	Chip Enable 2
RD	Output Enable
WR	Write Enable
VCC	Power Supply Voltage
VSS	Ground
n.c.	not connected

Q C

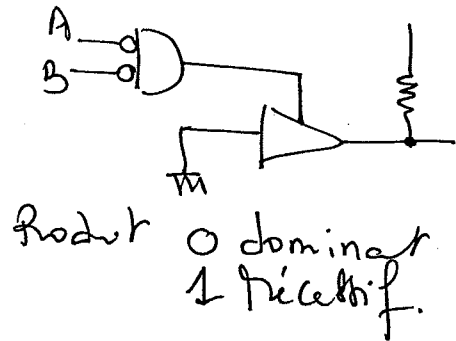
Corrigé ~~PAR~~ PARTIEL UE441.

A - niveau logique dominant est produit par un composant actif (transistor)

niveau récessif est produit par un composant passif (R)  
dominant plus fort que le récessif

B - soit le signal S

$$S_{oe} = \overline{(A \oplus B)};$$
$$S = 0;$$



Ex1:

1) C<sub>1</sub>: il pleut.

C<sub>2</sub>: Température est au-dessus de 24°C

on définit alors que:

3)  $P_1 = \overline{P_3} \cdot \overline{P_2}$

on effect P<sub>3</sub> implique que le père utilise la voiture.

P<sub>2</sub> implique que tania ira à la plage donc le contraire  $\Rightarrow$  elle est libre  $\Rightarrow$  elle l'accompagne.

4)

$$P_2 = \overline{C_1} \cdot C_2.$$

P<sub>3</sub>: va au théâtre  $\Rightarrow$  il utilise la voiture.

$$P_3 = C_1 + C_2.$$

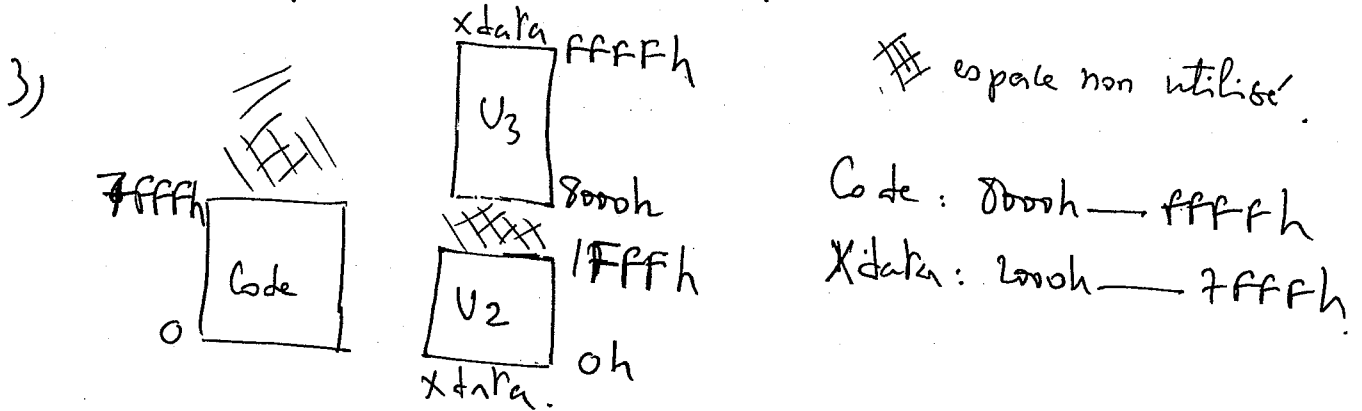
1) note  $P_1 = \overline{C_1} \cdot \overline{C_2}$

Michel ira au cinéma s'il ne pleut pas et si la température est  $\leq 24^\circ$ .

EX 3 :

1) U2 et U3 car signaux Cs (chipselect) non câblés.

2) Adresse de fin de U2 : 1FFFh.

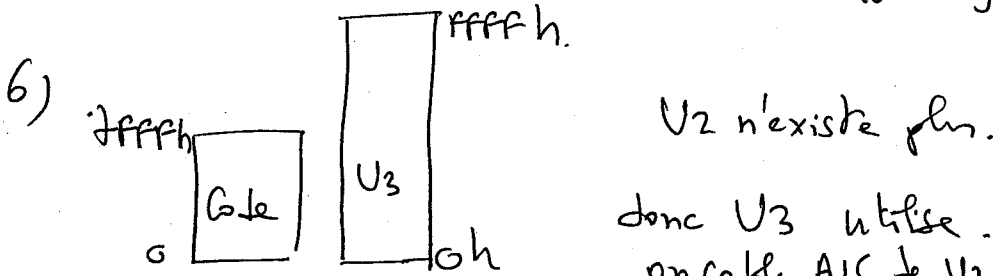


4) U2:  $Cs1 = A15 \# A14 \# A13$ ; actif niveau 0.

U3: OE = RD.

$Cs1 = !A15$ ;

5) U3 a 17 bits d'adresse (voir annexe)  $\Rightarrow 2^{17} \Rightarrow 128Ko$   
~~espace~~ espace non utilisé =  $128Ko - 32Ko = 96Ko$ .



donc U3 utilise 16 bits d'adresse.  
on cable AIS de U3 à AIS du  $\mu C$ .

7)  $Cs1$  du U3:  $Cs1 = 0$

ou alors.

$Cs1 = PSEN$

Car si  $PSEN = 1$  alors  
Pas accès Code  $\Rightarrow$   
accès mémoire.